



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/505,949	02/15/2000	Michael Chow	042390.P6447	5605

7590 04/18/2006

Thomas M Coester
Blakely Sokoloff Taylor and Zafman LLP
12400 Wilshire Boulevard Seventh Floor
Los Angeles, CA 90025

EXAMINER

LI, AIMEE J

ART UNIT

PAPER NUMBER

2183

DATE MAILED: 04/18/2006

Please find below and/or attached an Office communication concerning this application or proceeding.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

MAILED

MAR 18 2006

Technology Center 2100

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 09/505,949
Filing Date: February 15, 2000
Appellant(s): CHOW ET AL.

Joseph Lutz
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 14 July 2005 appealing from the Office action mailed 14 December 2004.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Graf, Rudolf F. Modern Dictionary of Electronics. Sixth Edition. Indianapolis, Indiana: Howard W. Sams & Company, ©1984. Pages 1132, term "word" and 1133, term "word size".

Art Unit: 2183

Heuring, Vincent P. and Jordan, Harry F. Computer Systems Design and Architecture.

Reading, Mass.: Addison-Wesley, ©1997. Pages 90-95, Section 3.2.

Memorandum from Stephen G. Kunin dated 25 June 2003

www.dictionary.com search term: token

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-5, 7-13, and 15-19 rejected under 35 U.S.C. 102(b) as being taught by

Blomgren et al., U.S. Patent Number 5,685,009 (herein referred to as '009), and U.S. Patent Number 5,781,750 (herein referred to as '750) incorporated by reference into '009 at column 3, lines 44-47. The shared registers disclosed in '009 are for use in the device disclosed in '750, as shown in '009 in column 1, line 24 to column 2, line 33; column 3, lines 444-67; and column 4, lines 6-16, hence '750 is incorporated by reference in '009 in the same embodiment.

Referring to claim 1, '009 and '750 have taught a processor comprising:

A first instruction set engine to process instructions from a first instruction set architecture (ISA) having a first word size ('009 column 1, lines 24-33 and 44-48 and column 1, line 59 to column 2, line 4; and '750 Abstract; column 3, lines 51-55 and 59-65; column 6, lines 16-24; and Figure 2). In regards to '009 and '750,

it is inherent that the two ISAs are different sizes, since RISC, specifically the x86 instruction set, and CISC, specifically the PowerPC instruction set, are different word sizes. For more information please see the provided information for more information on the x86 and PowerPC instruction sets and Heuring and Jordan's Computer Systems Design and Architecture provided with the action dated 18 June 2003 about the RISC and CISC instruction sets.

A second instruction set engine to process instructions from a second ISA having a second word size, the second word size being different than the first word size ('009 column 1, lines 24-33 and 44-48 and column 1, line 59 to column 2, line 4; and '750 Abstract; column 3, lines 51-55 and 59-65; column 6, lines 16-24; and Figure 2). In regards to '009 and '750, it is inherent that the two ISAs are different sizes, since RISC, specifically the x86 instruction set, and CISC, specifically the PowerPC instruction set, are different word sizes. For more information please see the provided information for more information on the x86 and PowerPC instruction sets and Heuring and Jordan's Computer Systems Design and Architecture provided with the action dated 18 June 2003 about the RISC and CISC instruction sets.

A mode identifier ('750 Abstract; column 3, line 65 to column 4, line 2; column 4, lines 7-11; column 6, lines 53-57; and Figure 2);

A plurality of floating-point registers shared by the first instruction set engine and the second instruction set engine ('009 column 2, lines 13-16; column 2, lines 41-

58; column 10, lines 43-53; column 18, lines 3-16; column 19, lines 47-48;
column 20, lines 10-15; Figures 6-8); and

A floating-point unit coupled to the floating-point registers, the floating-point unit processing an input responsive to the mode identifier to produce an output ('009 column 22, lines 48-54; and '750 column 3, line 65 to column 4, line 2; column 4, lines 7-11; column 6, lines 53-57; column 7, lines 1-12; and Figure 2).

Referring to claim 2, '009 and '750 have taught wherein the mode identifier is one of a plurality of bits in a processor status register ('750 Abstract; column 3, line 65 to column 4, line 2; column 4, lines 7-11; column 6, lines 53-57; and Figure 2).

Referring to claim 3, '009 and '750 have taught wherein the floating-point unit comprises:

Pre-processing hardware to detect if a token exists in the input ('009 column 1, lines 43-47 and column 2, lines 51-58; and '750 column 3, line 65 to column 4, line 21; column 6, lines 53-59; column 7, lines 30-54; and Figure 2);

An arithmetic unit responsive to the input and the mode identifier ('009 column 1, lines 43-47; and '750 column 6, line 53 to column 7, line 12; and Figure 2); and

Post-processing hardware to perform a token specific operation if a token exists in the input ('009 column 3, lines 57-67; and '750 column 5, line 49 to column 6, line 13).

Referring to claim 4, '009 and '750 have taught wherein the input includes data stored in at least one of the floating-point registers ('009 column 4, lines 17-34; column 9, lines 21-28; column 18, lines 3-16; column 19, lines 47-48; column 20, lines 10-15; and Figures 7-8).

Referring to claim 5, '009 and '750 have taught wherein the input may contain a token ('009 column 1, lines 43-47 and column 2, lines 51-58; and '750 column 3, line 65 to column 4, line 21; column 6, lines 53-59; column 7, lines 30-54; and Figure 2), wherein the floating-point registers are 82 bits wide ('009 column 2, lines 13-22), and wherein the token being an 82 bit processor known value ('009 column 1, lines 43-47 and column 2, lines 13-22 and 51-58; and '750 column 3, line 65 to column 4, line 21; column 6, lines 53-59; column 7, lines 30-54; and Figure 2).

Referring to claim 7, '009 and '750 have taught wherein the floating point registers each comprise

A sign bit ('009 column 2, lines 12-22). In regard to '009, it is inherent that the significand, also known as the mantissa, includes the sign bit. Please see FOLDOC definition mantissa ©1996 provided with the Office Action dated 30 June 2003.

An exponent ('009 column 2, lines 12-22); and

A significand ('009 column 2, lines 12-22). In regards to '009, it is inherent and well-known in the art that a significand is the same as the mantissa. Please see David Goldberg's "What Every Computer Scientist Should Know About Floating-point Arithmetic" ©1991, specifically under the section titled "Floating-point Formats", paragraph 2 "where $d.dd\dots d$ is called the *significand*" which refers to footnote 2 which says "This term was introduced by Forsythe and Moler [1967], and has generally replaced the older term *mantissa*."

Referring to claim 8, '009 and '750 have taught wherein the mode identifier indicates whether the processor is in a first mode or a second mode ('750 Abstract; column 3, line 65 to column 4, line 2; column 4, lines 7-11; column 6, lines 53-57; and Figure 2).

Referring to claim 9, '009 and '750 have taught wherein the mode identifier indicates whether the processor is in a 32 bit word ISA mode or a 64 bit word ISA mode ('009 column 1, lines 24-33 and 44-48 and column 1, line 59 to column 2, line 4; and '750 Abstract; column 3, lines 51-55 and 59-65; column 6, lines 16-24; and Figure 2). In regards to '009 and '750, the PowerPC instruction mode, i.e. RISC instruction mode, has 32 bit instruction words, as is the nature of the PowerPC instruction set. The x instruction mode, i.e. CISC instruction mode, has variable length instructions, which includes the 64-bit instruction length. For more information please see the provided information for more information on the x86 and PowerPC instruction sets and Heuring and Jordan's Computer Systems Design and Architecture provided with the action dated 18 June 2003 about the RISC and CISC instruction sets.

Referring to claim 10, '009 and '750 have taught a method in a processor comprising:

Fetching an input from at least one of a plurality of floating-point registers ('009 column 2, lines 13-16; column 2, lines 41-58; column 10, lines 43-53; column 18, lines 3-16; column 19, lines 47-48; column 20, lines 10-15; Figures 6-8);

Detecting whether the input includes a token ('009 column 1, lines 43-47 and column 2, lines 51-58; and '750 column 3, line 65 to column 4, line 21; column 6, lines 53-59; column 7, lines 30-54; and Figure 2);

If the token is detected in the input, checking what mode the processor is in ('009 column 1, lines 43-47 and column 2, lines 51-58; and '750 column 3, line 65 to column 4, line 21; column 6, lines 53-59; column 7, lines 30-54; and Figure 2);

If the processor is in a first mode, processing the input to render an arithmetic result ('009 column 1, lines 43-47; and '750 column 6, line 53 to column 7, line 12; and Figure 2);

If the processor is in a second mode, performing a token specific operation ('009 column 3, lines 57-67; and '750 column 5, line 49 to column 6, line 13); and
Producing an output ('009 column 1, lines 43-47 and column 3, lines 57-67; and '750 column 5, line 49 to column 6, line 13; column 6, line 53 to column 7, line 12; and Figure 2).

Referring to claim 11, '009 and '750 has taught

Wherein the input is comprised of at least one operand and at least one operator ('750 column 3, lines 51-56). In regards to '750, the PowerPC RISC instruction set and x86 CISC instruction set both have at least one operand and at least one operator. Please see the provided information on the PowerPC and x86 instruction sets.

Wherein detecting comprises examining the at least one operand to determine whether any of the operands correspond to the token ('009 column 1, lines 43-47 and column 2, lines 51-58; and '750 column 3, line 65 to column 4, line 21; column 6, lines 53-59; column 7, lines 30-54; and Figure 2); and

Wherein checking comprises examining a mode identifier to determine whether the processor is in the first mode or the second mode ('009 column 1, lines 43-47 and column 2, lines 51-58; and '750 column 3, line 65 to column 4, line 21; column 6, lines 53-59; column 7, lines 30-54; and Figure 2).

Referring to claim 12, '009 and '750 have taught wherein processing comprises executing at least one operation on the at least one operand according to the at least one operator to achieve a result ('009 column 1, lines 43-47; and '750 column 6, line 53 to column 7, line 12; and Figure 2).

Referring to claim 13, '009 and '750 have taught wherein performing comprises propagating the token ('009 column 3, lines 57-67; and '750 column 5, line 49 to column 6, line 13); and wherein producing output comprises setting the output to be the token ('009 column 3, lines 57-67; and '750 column 5, line 49 to column 6, line 13). In regards to '009 and '750, the indication that the CPU is in emulation mode must be propagated through the entire process of switching from CISC to RISC and back to CISC.

Referring to claim 15, '009 and '750 have taught wherein checking comprises checking a mode identifier ('009 column 1, lines 43-47 and column 2, lines 51-58; and '750 column 3, line 65 to column 4, line 21; column 6, lines 53-59; column 7, lines 30-54; and Figure 2).

Referring to claim 16, '009 and '750 have taught wherein checking comprises checking a mode identifier bit in a processor status register ('009 column 1, lines 43-47 and column 2, lines 51-58; and '750 column 3, line 65 to column 4, line 21; column 6, lines 53-59; column 7, lines 30-54; and Figure 2).

Referring to claim 17, '009 and '750 have taught wherein the first mode is a 32 bit word ISA mode and the second mode is a 64 bit word ISA mode ('009 column 1, lines 24-33 and 44-48 and column 1, line 59 to column 2, line 4; and '750 Abstract; column 3, lines 51-55 and 59-65; column 6, lines 16-24; and Figure 2). In regards to '009 and '750, the PowerPC instruction mode, i.e. RISC instruction mode, has 32 bit instruction words, as is the nature of the PowerPC instruction set. The x instruction mode, i.e. CISC instruction mode, has variable length instructions, which includes the 64-bit instruction length. For more information please see the provided information for more information on the x86 and PowerPC instruction sets and Heuring and Jordan's Computer Systems Design and Architecture provided with the action dated 18 June 2003 about the RISC and CISC instruction sets.

Referring to claim 19, '009 and '750 have taught a method in a multi-mode processor comprising:

Fetching an input from at least one of a plurality of floating-point registers ('009 column 2, lines 13-16; column 2, lines 41-58; column 10, lines 43-53; column 18, lines 3-16; column 19, lines 47-48; column 20, lines 10-15; Figures 6-8);

Detecting whether the input includes at least one token of a plurality of tokens ('009 column 1, lines 43-47 and column 2, lines 51-58; and '750 column 3, line 65 to column 4, line 21; column 6, lines 53-59; column 7, lines 30-54; and Figure 2);

If at least one token is detected in the input, checking what mode the processor is in ('009 column 1, lines 43-47 and column 2, lines 51-58; and '750 column 3, line

Art Unit: 2183

65 to column 4, line 21; column 6, lines 53-59; column 7, lines 30-54; and Figure 2);

Processing the input to render an arithmetic result when the processor is in at least a first mode of a plurality of modes ('009 column 1, lines 43-47; and '750 column 6, line 53 to column 7, line 12; and Figure 2); and

Performing a token specific operation when the processor is in at least a second mode of a plurality of modes ('009 column 3, lines 57-67; and '750 column 5, line 49 to column 6, line 13).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 6 and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Blomgren et al., U.S. Patent Number 5,685,009 (herein referred to as '009), and U.S. Patent Number 5,781,750 (herein referred to as '750) incorporated by reference into '009 at column 3, lines 44-47 in view of InstantWeb's Online Computing Dictionary terms "speculative evaluation" and "speculative execution" (herein referred to as FOLDLOC). '009 and '750 have taught wherein the token represents a "not a thing value" (NaTVal) that defines an unsuccessful load request ('750 column 4, lines 13-33 and column 7, lines 25-32). In regards to '750, the TLB is used to load the physical address of the instruction when virtual addresses are present and emulation mode is entered when a miss occurs in the TLB, i.e. cannot find the physical address

Art Unit: 2183

to load, thereby creating an unsuccessful load request. FOLDOC has taught speculative evaluation and execution of instructions (FOLDOC terms “speculative evaluation” and “speculative execution”). A person of ordinary skill in the art at the time the invention was made would have recognized that speculative evaluation and execution reduces the overall run-time of a process and keeps all functional units working, i.e. not wasted cycles, (FOLDOC terms “speculative evaluation” and “speculative execution”), thereby increasing processor speed and efficiency. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the speculative evaluation and execution of FOLDOC in the device of ‘009 and ‘750 to improve processor speed and efficiency.

Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over Blomgren et al., U.S. Patent Number 5,685,009 (herein referred to as ‘009), and U.S. Patent Number 5,781,750 (herein referred to as ‘750) incorporated by reference into ‘009 at column 3, lines 44-47 in view of Amos Omondi’s The Microarchitecture of Pipelined and Superscalar Computers ©1999 (herein referred to as Omondi). ‘009 and ‘750 have taught a multi-mode processor comprising:

A plurality of instruction set engines to process instructions from a plurality of instruction set architectures having different word sizes (‘009 column 1, lines 24-33 and 44-48 and column 1, line 59 to column 2, line 4; and ‘750 Abstract; column 3, lines 51-55 and 59-65; column 6, lines 16-24; and Figure 2). In regards to ‘009 and ‘750, it is inherent that the two ISAs are different sizes, since RISC, specifically the x86 instruction set, and CISC, specifically the PowerPC instruction set, are different word sizes. For more information please see the provided information for more information on the x86 and PowerPC instruction

sets and Heuring and Jordan's Computer Systems Design and Architecture

provided with the action dated 18 June 2003 about the RISC and CISC instruction sets.;

A mode identifier ('750 Abstract; column 3, line 65 to column 4, line 2; column 4, lines 7-11; column 6, lines 53-57; and Figure 2);

A plurality of floating-point registers shared by the instruction set engines ('009 column 2, lines 13-16; column 2, lines 41-58; column 10, lines 43-53; column 18, lines 3-16; column 19, lines 47-48; column 20, lines 10-15; Figures 6-8); and

A floating-point unit coupled to the floating-point registers, the floating-point units processing an input responsive to the mode identifier ('009 column 22, lines 48-54; and '750 column 3, line 65 to column 4, line 2; column 4, lines 7-11; column 6, lines 53-57; column 7, lines 1-12; and Figure 2).

'009 and '750 have not explicitly taught a plurality of floating-point units. Omondi has taught a plurality of floating-point units (Omondi Figure 1.9; Figure 1.11; Figure 1.12; and Figure 1.13). A person of ordinary skill in the art at the time the invention was made would have recognized that a superscalar processor, such as those shown in Omondi, execute more than one instruction at a time, thereby increasing the speed and efficiency of a processor (Omondi page 6, section 1.3). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the superscalar processor of Omondi in the device of '009 and '750 to increase the speed and efficiency of the processor. Also, merely duplicating a part of a device for multiple effect is not a patentable difference, see *In re Harza*, 274 F.2d 669, 124 USPQ 378 (CCPA 1960).

The following is a table summarizing the rejections in question:

Instant Application's Claim	Prior Art Citation and Explanation
Claim 1	
A processor comprising:	
A first instruction set engine to process instructions from a first instruction set architecture (ISA) having a first word size;	<p>'009 column 1, lines 24-33 and 44-48 column 1, line 59 to column 2, line 4</p> <p>'750 Abstract column 3, lines 51-55 and 59-65 column 6, lines 16-24 Figure 2</p> <p>In regards to '009 and '750, it is inherent that the two ISAs are different sizes, since RISC, specifically the x86 instruction set, and CISC, specifically the PowerPC instruction set, are different word sizes. For more information please see the provided information for more information on the x86 and PowerPC instruction sets and Heuring and Jordan's <u>Computer Systems Design and Architecture</u> provided with the action dated 18 June 2003 about the RISC and CISC instruction sets.</p> <p>The term word size, as taught by Graf's <u>Modern Dictionary of Electronics</u> 6th Edition ©1984, is "the</p>

	<p>number of...binary bits comprising a word.” A word is</p> <p>“The number of bits needed to present <i>a computer instruction</i>, or the number of bits needed to represent the largest data element... (emphasis added)”</p>
<p>A second instruction set engine to process instructions from a second ISA having a second word size, the second word size being difference than the first word size;</p>	<p>‘009 column 1, lines 24-33 and 44-48 column 1, line 59 to column 2, line 4</p> <p>‘750 Abstract column 3, lines 51-55 and 59-65 column 6, lines 16-24 Figure 2</p> <p>In regards to ‘009 and ‘750, it is inherent that the two ISAs are different sizes, since RISC, specifically the x86 instruction set, and CISC, specifically the PowerPC instruction set, are different word sizes. For more information please see the provided information for more information on the x86 and PowerPC instruction sets and Heuring and Jordan’s <u>Computer Systems Design and Architecture</u> provided with the action dated 18 June 2003 about the RISC and CISC instruction sets.</p> <p>The term word size, as taught by Graf’s <u>Modern Dictionary of Electronics</u> 6th Edition ©1984, is “the</p>

	number of...binary bits comprising a word.” A word is “The number of bits needed to present <i>a computer instruction</i> , or the number of bits needed to represent the largest data element... (emphasis added)”
A mode identifier;	‘750 Abstract column 3, line 65 to column 4, line 2 column 4, lines 7-11 column 6, lines 53-57 Figure 2
A plurality of floating-point registers shared by the first instruction set engine and the second instruction set engine; and	‘009 column 2, lines 13-16 column 2, lines 41-58 column 10, lines 43-53 column 18, lines 3-16 column 19, lines 47-48 column 20, lines 10-15 Figure 6 Figure 7 Figure 8
A floating-point unit coupled to the floating-point registers, the floating-point unit processing an input responsive to the mode identifier to produce an output.	‘009 column 22, lines 48-54 ‘750 column 3, line 65 to column 4, line 2 column 4, lines 7-11 column 6, lines 53-57 column 7, lines 1-12 Figure 2
Claim 3	
Wherein the floating-point unit comprises:	

Pre-processing hardware to detect if a token exists in the input;	'009 column 1, lines 43-47 column 2, lines 51-58 '750 column 3, line 65 to column 4, line 21 column 6, lines 53-59 column 7, lines 30-54 Figure 2
An arithmetic unit responsive to the input and the mode identifier; and	'009 column 1, lines 43-47 '750 column 6, line 53 to column 7, line 12 Figure 2
Post-processing hardware to perform a token specific operation if a token exists in the input.	'009 column 3, lines 57-67 '750 column 5, line 49 to column 6, line 13
Claim 6	
Wherein the token represents a "not a thing value" (NaTVal) that defines an unsuccessful load request	'750 column 4, lines 13-33 column 7, lines 25-32 In regards to '750, the TLB is used to load the physical address of the instruction when virtual addresses are present and emulation mode is entered when a miss occurs in the TLB, i.e. cannot find the physical address to load, thereby creating an unsuccessful load request.
Speculative	FOLDOC "speculative evaluation" "speculative execution"

	<p>A person of ordinary skill in the art at the time the invention was made would have recognized that speculative evaluation and execution reduces the overall run-time of a process and keeps all functional units working, i.e. not wasted cycles, (FOLDLOC terms “speculative evaluation” and “speculative execution”), thereby increasing processor speed and efficiency.</p> <p>Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the speculative evaluation and execution of FOLDLOC in the device of ‘009 and ‘750 to improve processor speed and efficiency.</p>
Claim 9	
<p>Wherein the mode identifier indicates whether the processor is in a 32 bit word ISA mode or a 64 bit word ISA mode</p>	<p>‘009 column 1, lines 24-33 and 44-48 column 1, line 59 to column 2, line 4 ‘750 Abstract column 3, lines 51-55 and 59-65 column 6, lines 16-24 Figure 2</p> <p>In regards to ‘009 and ‘750, the PowerPC instruction mode, i.e. RISC instruction mode, has 32 bit instruction words, as is the nature of the PowerPC instruction set.</p> <p>The x instruction mode, i.e. CISC instruction mode, has</p>

	<p>variable length instructions, which includes the 64-bit instruction length. For more information please see the provided information for more information on the x86 and PowerPC instruction sets and Heuring and Jordan's <u>Computer Systems Design and Architecture</u> provided with the action dated 18 June 2003 about the RISC and CISC instruction sets.</p> <p>The term word size, as taught by Graf's <u>Modern Dictionary of Electronics</u> 6th Edition ©1984, is "the number of...binary bits comprising a word." A word is "The number of bits needed to present <i>a computer instruction</i>, or the number of bits needed to represent the largest data element... (emphasis added)"</p>
Claim 10	
A method in a processor comprising:	
Fetching an input from at least one of a plurality of floating-point registers;	<p>'009</p> <p>column 2, lines 13-16 column 2, lines 41-58 column 10, lines 43-53 column 18, lines 3-16 column 19, lines 47-48 column 20, lines 10-15 Figure 6 Figure 7 Figure 8</p>

Art Unit: 2183

Detecting whether the input includes a token;	'009 column 1, lines 43-47 column 2, lines 51-58 '750 column 3, line 65 to column 4, line 21 column 6, lines 53-59 column 7, lines 30-54 Figure 2
If the token is detected in the input, checking what mode the processor is in;	'009 column 1, lines 43-47 column 2, lines 51-58 '750 column 3, line 65 to column 4, line 21 column 6, lines 53-59 column 7, lines 30-54 Figure 2
If the processor is in a first mode, processing the input to render an arithmetic result;	'009 column 1, lines 43-47 '750 column 6, line 53 to column 7, line 12 Figure 2
If the processor is in a second mode, performing a token specific operation; and	'009 column 3, lines 57-67 '750 column 5, line 49 to column 6, line 13
Producing an output.	'009 column 1, lines 43-47 column 3, lines 57-67 '750 column 5, line 49 to column 6, line 13 column 6, line 53 to column 7, line 12 Figure 2
Claim 18	
A multi-mode processor comprising:	
A plurality of instruction set	'009 column 1, lines 24-33 and 44-48 column 1, line 59 to column 2, line 4

engines to process instructions from a plurality of instruction set architectures having different word sizes;	<p>'750</p> <p>Abstract column 3, lines 51-55 and 59-65 column 6, lines 16-24 Figure 2</p> <p>In regards to '009 and '750, it is inherent that the two ISAs are different sizes, since RISC, specifically the x86 instruction set, and CISC, specifically the PowerPC instruction set, are different word sizes. For more information please see the provided information for more information on the x86 and PowerPC instruction sets and Heuring and Jordan's <u>Computer Systems Design and Architecture</u> provided with the action dated 18 June 2003 about the RISC and CISC instruction sets.</p> <p>The term word size, as taught by Graf's <u>Modern Dictionary of Electronics</u> 6th Edition ©1984, is "the number of...binary bits comprising a word." A word is "The number of bits needed to present <i>a computer instruction</i>, or the number of bits needed to represent the largest data element... (emphasis added)"</p>
A mode identifier;	<p>'750</p> <p>Abstract column 3, line 65 to column 4, line 2</p>

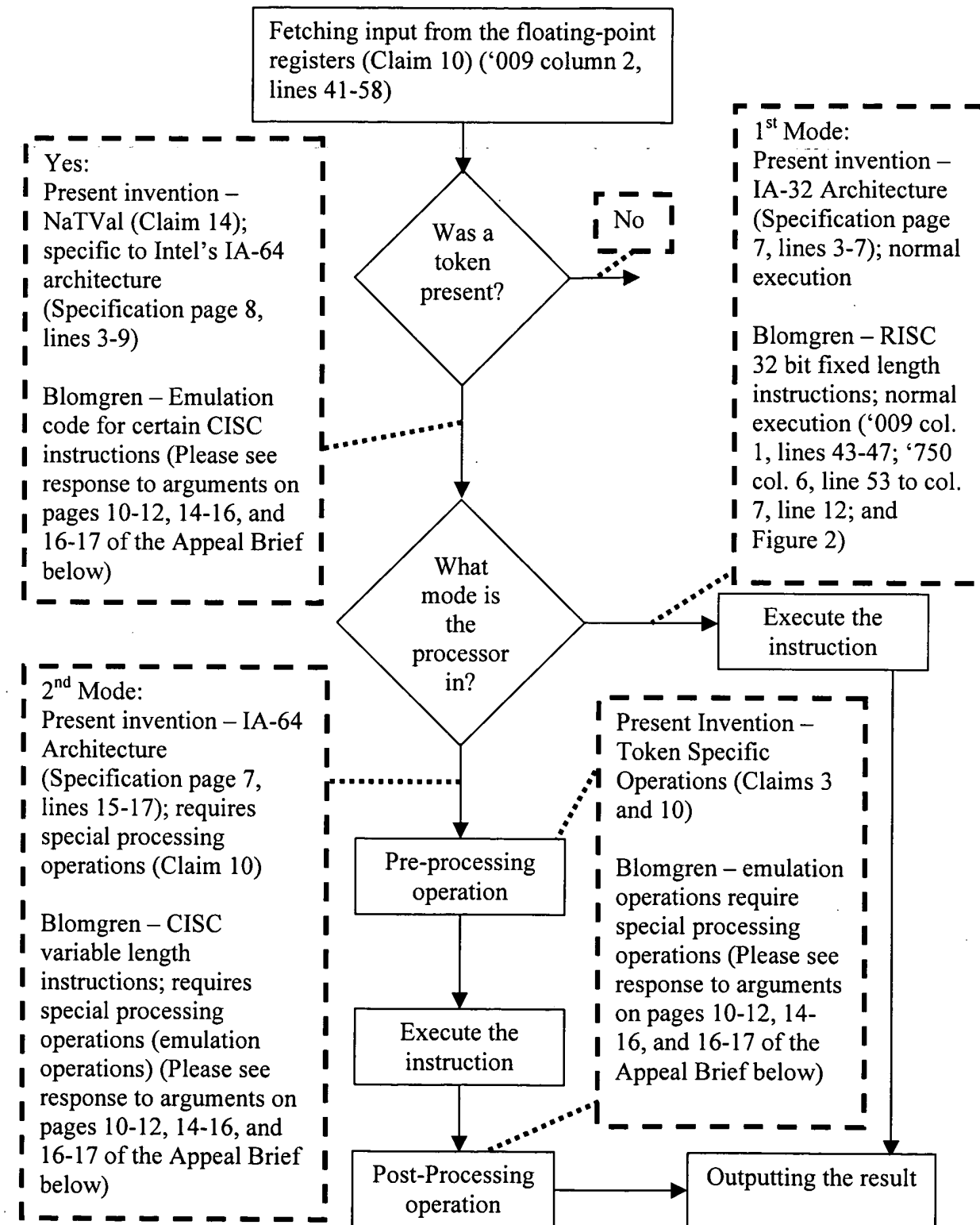
	<p>column 4, lines 7-11 column 6, lines 53-57 Figure 2</p>
<p>A plurality of floating-point registers shared by the instruction set engines; and</p>	<p>‘009 column 2, lines 13-16 column 2, lines 41-58 column 10, lines 43-53 column 18, lines 3-16 column 19, lines 47-48 column 20, lines 10-15 Figure 6 Figure 7 Figure 8</p>
<p>A floating-point unit coupled to the floating-point registers, the floating-point units processing an input responsive to the mode identifier.</p>	<p>‘009 column 22, lines 48-54 ‘750 column 3, line 65 to column 4, line 2 column 4, lines 7-11 column 6, lines 53-57; column 7, lines 1-12 Figure 2</p>
<p>A plurality of floating-point units</p>	<p>Omondi Figure 1.9 Figure 1.11 Figure 1.12 Figure 1.13</p> <p>A person of ordinary skill in the art at the time the invention was made would have recognized that a superscalar processor, such as those shown in Omondi, execute more than one instruction at a time, thereby increasing the speed and efficiency of a processor (Omondi page 6, section 1.3). Therefore, it would have</p>

	<p>been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the superscalar processor of Omondi in the device of '009 and '750 to increase the speed and efficiency of the processor. Also, merely duplicating a part of a device for multiple effect is not a patentable difference, see <i>In re Harza</i>, 274 F.2d 669, 124 USPQ 378 (CCPA 1960).</p>
Claim 19	
A method in a multi-mode processor comprising:	
Fetching an input from at least one of a plurality of floating-point registers;	<p>'009 column 2, lines 13-16 column 2, lines 41-58 column 10, lines 43-53 column 18, lines 3-16 column 19, lines 47-48 column 20, lines 10-15 Figure 6 Figure 7 Figure 8</p>
Detecting whether the input includes at least one token of a plurality of tokens;	<p>'009 column 1, lines 43-47 column 2, lines 51-58 '750 column 3, line 65 to column 4, line 21 column 6, lines 53-59 column 7, lines 30-54 Figure 2</p>
If at least one token is detected in the input, checking what mode the processor is in;	<p>'009 column 1, lines 43-47 column 2, lines 51-58 '750 column 3, line 65 to column 4, line 21 column 6, lines 53-59</p>

	column 7, lines 30-54 Figure 2
Processing the input to render an arithmetic result when the processor is in at least a first mode of a plurality of modes; and	'009 column 1, lines 43-47 '750 column 6, line 53 to column 7, line 12 Figure 2
Performing a token specific operation when the processor is in at least a second mode of a plurality of modes.	'009 column 3, lines 57-67 '750 column 5, line 49 to column 6, line 13

The following is a flow chart summarizing what the Examiner believes is recited in the claim language, with some additional information taken from the instant application's specification, and an explanation of Blomgren in relation to certain aspects of the flow chart.

Art Unit: 2183



(10) Response to Argument

Applicant argues in essence on pages 7-10, 12-14, and 19-20

Hence, Applicants submit that the entire specification of Blomgren and Blomgren2 are devoid of any reference to providing processing for different ISAs having different ISA word sizes. Applicant's submit that the explicit text of Blomgren (*See*, col. 1, lines 44-56) and Blomgren2 (*See*, col. 9, lines 16-20) imply that the instruction word size is the same for both the CISC and RISC instruction sets, and in accordance with the definition of "word size" provide above.

Yet, in spite of the lack of any teaching toward processing of ISAs having a different word size, by improperly equating the term "instruction size" with the term "word size", the Examiner incorrectly finds a teaching within Blomgren and Blomgren2 to anticipate independent claims 1 and 18...

This has not been found persuasive. The claim language states, taking claim 1 as exemplary, "a first instruction set engine to process instructions from a first instruction set architecture (ISA) having a first word size; a second instruction set engine to process instructions from a second ISA having a second word size, the second word size being different than the first word size...". According to the claim language, the claim can be interpreted two separate ways. The first interpretation, as shown in Diagram 1 below, has the language of "a first instruction set architecture (ISA)" modifying the instructions being processed by a first instruction set engine and the "having a first word size" modifying the first ISA. The second interpretation, as shown in Diagram 2 below, has both the language of "a first instruction set architecture (ISA)" and "having a first word size" modifying the instructions being processed by a first instruction set engine. For the rejection in question, the Examiner interpreted the claims according to Diagram 2 with the ISA and word size modifying the instructions being processed. The same interpretation was taken for similar limitations and claim language.

Diagram 1

A first instruction set engine to process instructions

from a first instruction set architecture (ISA)

having a first word size

Diagram 2

A first instruction set engine to process instructions

from a first instruction set architecture (ISA)

having a first word size

Applicant's arguments seem to be arguing the first interpretation of the claims, with the word size modifying the ISA, which requires that the data size for each instruction engine be completely separate. However, this interpretation is not clear by the claim language. Applicant tries to further prove this stance by providing the definition for "word size" provided in Applicant's Exhibit A. However, the Examiner could not locate this explicit definition for "word size" in the specification. There were insinuations of this definition but no definition was "set out...explicitly and with 'reasonable clarity, deliberateness, and precision' in the disclosure to give one of ordinary skill in the art notice of the change (Memorandum from Stephen G. Kunin dated 25 June 2003)." As stated in the table above with explanation of the rejection, the Examiner relied upon Graf for the definition of "word size". Specifically, the term word size, as taught by Graf's Modern Dictionary of Electronics 6th Edition ©1984, is "the number of...binary bits comprising a word." A word, according to the third definition in Graf, is "The number of bits needed to present *a computer instruction*, or the number of bits needed to represent the

Art Unit: 2183

largest data element... (emphasis added)” According to this definition, a “word” is considered the number of bits to represent a computer instruction or data. Since the definition is in the alternative, the Examiner interpreted the claim language with a “word” being “The number of bits needed to present a computer instruction” and a “word size” being “the number of...binary bits comprising a word”, e.g. the number of binary bits comprising a computer instruction. Therefore, interpreting the CISC and RISC instructions, which inherently have different instruction sizes, as shown in the rejection table above, have different word sizes. ‘009 and ‘750 (referred to as Blomgren and Blomgren2 in Applicant’s arguments) teach this aspect. ‘009 teaches in lines 24-33 and 44-48 and column 1, line 59 to column 2, line 4 a device that executes both CISC and RISC instructions. This means that the device stores, fetches, and executes RISC and CISC instructions. This is reiterated in ‘750 Abstract; column 3, lines 51-55 and 59-65; and column 6, lines 16-24. The Examiner provided with the action dated 18 June 2003 Heuring and Jordan’s Computer Systems Design and Architecture to show that RISC and CISC instructions are different instruction sizes, e.g. their storage word sizes are different. The device of ‘009 and ‘750 fetches both of these instructions from storage, e.g. fetches instructions of different word sizes, decodes them, and executes them. In ‘750, RISC and CISC instructions follow different paths using similar hardware but having different results. The main difference in the paths is in relation to the decoder, which is different. For example, RISC instructions in Figure 2 travel from fetching element 32 to instruction identifier element 36 (RISC ID) to execution unit element 48. That is the instruction set engine for a RISC instruction. CISC instructions in Figure 2 travel from fetching element 32 to instruction identifier element 36 (CISC ID or Emu ID) to execution unit element 48. That is the instruction set engine for a CISC instruction. An

Art Unit: 2183

instruction set engine, according to the claim, is merely something that processes instructions from a specific instruction set. The paths set forth in Blomgren process each specific instruction set, specifically, there is specific decoder logic related to RISC and CISC instructions.

Applicant's arguments suggest that the definition of "word size" is narrower than the actual definition of "word size". However, barring an explicit definition "with 'reasonable clarity, deliberateness, and precision' in the disclosure to give one of ordinary skill in the art notice of the change...it would **not** be proper for the examiner to give words of the claim special meaning when no such special meaning has been defined by the applicant in the written description (Memorandum from Stephen G. Kunin dated 25 June 2003)." In addition, Applicant's arguments seem to insinuate that there are separate, individual instruction set engines for each word size, but there is no language in the claim to eliminate instruction set engines that share elements, such as those in Blomgren. The arguments also seem to insinuate that the instruction set engines must have different bus sizes, as set forth in Applicant's Exhibit A definition of "word size". However, the definition from Applicant's Exhibit A uses the wording "often", which suggests that the word size must always match the bus size is not always true, as is shown in Blomgren's device.

Applicant argues in essence on pages 10-12 and 14-16

Accordingly, although Blomgren and Blomgren2 describe an emulation mode to emulate the unknown instruction, the processing performed within Blomgren and Blomgren2 does not vary according to a processor mode as indicated by a mode identifier. Applicant respectfully submit that the detection of an unknown opcode and entry into emulation to emulate an unknown instruction does not teach the detection of whether an input contains a token or performing a token-specific operation according to a processor mode...

This has not been found persuasive. The claim language states, taking claim 3 as exemplary, “pre-processing hardware to detect of a token exists in the input; an arithmetic unit responsive to the input and the mode identifier; and post-processing hardware to perform a token specific operation if a token exists in the input.” ‘750 teaches these aspects. ‘750 teaches in column 3, line 65 to column 4, line 21; column 6, lines 53-59; and column 7, lines 30-54; and Figure 2 that the “unknown instruction” signal is sent to the mode control unit, and from there to the mode register and multiplexer for choosing the correct decoder. The “unknown instruction” is the signal showing whether a token, which is merely “something serving as an indication, proof, or expression of something else (dictionary.com)”, is present. The token, in ‘750, is whether the device is in CISC instruction emulation mode or not. The execution unit in ‘750’s Figure 2, element 48 executes the decoded instruction from multiplexer 46, so it is responsive to the input and mode identifier, since it executes the instruction output from the decoders that received the input instruction and the multiplexer that selects the correctly decoded instruction in response to the unknown instruction signal 40, mode control unit 42, and mode register 38. With regards to the post-processing hardware, there are two separate interpretations of this language. The first interpretation is that this refers to hardware after the decode processing. ‘750 shows this in column 7, lines 1-12, which essentially states that, when in emulation mode, the post-processing hardware specifically executes microcode stored in emulation memory as opposed to the decoded operations from the RISC and CISC decoders. The second interpretation is that the post-processing hardware refers to hardware performing an operation after execution of the emulated instruction is complete. ‘750 also teaches this in column 7, lines 26-29, which states,

Art Unit: 2183

essentially, that emulation mode always uses loads the TLB so that the emulation driver has “the highest level of control over address mapping and translation.”

Applicant argues in essence on pages 16-17

It is well established that obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention absent the teaching or suggestion supporting such combination...

...

Accordingly, although Blomgren and Blomgren2 describe an emulation mode to emulate the unknown instruction, the processing performed within Blomgren and Blomgren2 does not vary according to a processor mode. Applicant respectfully submit that the detection of an unknown opcode and entry into emulation to emulate an unknown instruction does not teach the detection of whether an input contains a token or a token-specific operation according to a processor mode...

This has not been found persuasive. The claim language states “wherein the token represent a ‘not a thing value’ (NaTVal) that defines an unsuccessful speculative load request.” As stated above, the “unknown instruction” is the signal showing whether a token, which is merely “something serving as an indication, proof, or expression of something else (dictionary.com)”, is present. The token, in ‘750, is whether the device is in emulation mode or not. ‘750 states in column 7, lines 30-31 that whenever the TLB misses, the device enters emulation mode. ‘750 teaches in column 7, lines 12-16 that the TLB translates the addresses sent to it by the execution unit from virtual addresses to physical addresses. Hence, when an address in the TLB cannot be translated, e.g. a load miss, then emulation mode is entered. As shown in the rejection above, ‘750 has not taught that the load is speculative. The Examiner relied upon InstantWeb’s FOLDOC to show this in “speculative evaluation” and “speculative execution”. Applicant’s argument states that they do not believe the motivation to combine and the Examiner used improper hindsight. However, “speculative evaluation” explicitly states “it can reduce the overall run-time by making some needed results available earlier

Art Unit: 2183

than they would be otherwise” and “allows a superscalar processor to keep its functional units as busy as possible” which is the motivation cited by the Examiner. These explicit statements show the motivation and show that this knowledge was available to a person of ordinary skill in the art at the time the invention was made, thereby obviating the impermissible hindsight argument. Reducing overall run-time is improving processor speed, since processor speed is dependent upon how much time it takes for a program to run on a processor. If a processor takes less time to execute a program, i.e. it executes a program faster, then the speed of the processor has increased. Keeping functional units as busy as possible is improving processor efficiency since idle cycles in functional units are reduced. If a processor uses its resources with minimized periods where no work is being done by the resource, the efficiency of the processor is improved.

(11) Related Proceeding(s) Appendix

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Aimee J. Li

Conferees:

Eddie Chan



EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Donald Sparks



DONALD SPARKS
SUPERVISORY PATENT EXAMINER